WHAT IS CLAIMED IS:

1. A method for load balancing code execution, said method comprising:

retrieving a code segment from a plurality of code segments;

identifying a processor type from a plurality of processor types to execute the code segment; and

loading the code segment into a processor that corresponds to the identified processor type.

2. The method as described in claim 1 wherein the code segment is byte code.

3. The method as described in claim 2 wherein the byte code includes a byte code type, the byte code type selected from the group consisting of Java, XML, HTML, Shader, and Script.

4. The method as described in claim 2 further comprising:

compiling source code, the compiling resulting in the byte code;

determining whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code;

storing the pointer in the byte code file in response to the determination;

storing the byte code at the stored location in response to the determination; and

performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location

and retrieving the byte code in response to the analyzing.

5. The method as described in claim 1 wherein the identifying includes analyzing the availability of each of the plurality of processor types, and wherein the analyzing includes retrieving a loading factor for each of the plurality of processor types.

6. The method as described in claim 1 wherein the identifying further comprises:
detecting one or more operations included in the code segment; and

matching one or more of the operations with one of the processor types from the plurality of processor types.

7. The method as described in claim 1 wherein the identifying further comprises:
determining whether the code segment includes a program directive corresponding to one of the plurality of processors; and

matching one or more of the operations with one of the processor types from the plurality of processor types in response to the determination.

8. An information handling system comprising:
a plurality of processors;

a memory accessible by the processors;

one or more nonvolatile storage devices accessible by the processors; and

a code execution load balancing tool for load balancing code execution, the code execution load balancing tool comprising software code effective to:

retrieve a code segment from a plurality of code segments located in the memory;

identify a processor type from a plurality of processor types to execute the code segment; and

load the code segment into one of the processors from the plurality of processors that corresponds to the identified processor type.

9. The information handling system as described in claim 8 wherein the code segment is byte code.

10. The information handling system as described in claim 9 wherein the byte code includes a byte code type, the byte code type selected from the group consisting of Java, XML, HTML, Shader, and Script.

11. The information handling system as described in claim 9 wherein the software code is further effective to:
compile source code, the compiling resulting in the byte code;

determine whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code;

store the pointer in the byte code file in the memory in response to the determination;

store the byte code at the stored location in the memory in response to the determination; and

perform the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code from the memory in response to the analyzing.

12. The information handling system as described in claim 8 wherein the identifying includes analyzing the availability of each of the plurality of processor types, and wherein the analyzing includes retrieving a loading factor for each of the plurality of processor types.

13. The information handling system as described in claim 8 wherein the software code is further effective to:
detect one or more operations included in the code segment; and

match one or more of the operations with one of the processor types from the plurality of processor types.

14. A computer program product stored on a computer operable media for load balancing code execution, said computer program product comprising:
means for retrieving a code segment from a plurality of code segments;

means for identifying a processor type from a plurality of processor types to execute the code segment; and

means for loading the code segment into a processor that corresponds to the identified processor type.

15. The computer program product as described in claim 14 wherein the code segment is byte code.

16. The computer program product as described in claim 15 wherein the byte code includes a byte code type, the byte code type selected from the group consisting of Java, XML, HTML, Shader, and Script.

17. The computer program product as described in claim 15 further comprising:
means for compiling source code, the compiling resulting in the byte code;

means for determining whether to store a pointer in a byte code file, the pointer including a stored location that corresponds to the byte code;

means for storing the pointer in the byte code file in response to the determination;

means for storing the byte code at the stored location in response to the determination; and

means for performing the retrieving using the pointer, wherein the retrieving includes analyzing the stored location and retrieving the byte code in response to the analyzing.

18. The computer program product as described in claim 14 wherein the identifying includes analyzing the availability of each of the plurality of processor types, and wherein the analyzing includes retrieving a

loading factor for each of the plurality of processor types.

19. The computer program product as described in claim 14 wherein the identifying further comprises:

means for detecting one or more operations included in the code segment; and

means for matching one or more of the operations with one of the processor types from the plurality of processor types.

20. The computer program product as described in claim 14 wherein the identifying further comprises:

means for determining whether the code segment includes a program directive corresponding to one of the plurality of processors; and

means for matching one or more of the operations with one of the processor types from the plurality of processor types in response to the determination.